

parameter box for MIDI software

convenient software synthesizer operation

Using a keyboard and mouse to operate the innumerable knobs and sliders of a PC mixing board can turn playing around with a sound synthesizer into an agonizing experience. With the MIDI parameter box it's a different story!



Almost every PC these days has a sound card. However, most PC users employ it only to reproduce operating system sounds, music from audio CDs and sound effects for games and presentations. They thus use only the standard features of the software for the sound card, even though there are outstanding programs available that allow even non-musicians to exploit the manifold features of modern sound cards. These are sound synthesizer programs.

Such synthesizers are sometimes even included in the software packages that come with the better quality sound cards. However, there are also separate programs, such as *Generator* from Native Instruments or *Rebirth* from Propellerhead. These programs simulate the functions and operations of a real synthesizer using a screen full of sliders and knobs that are controlled by the mouse. They represent a sort of Gordian knot for anyone who wants to do more than adjust a single slider.

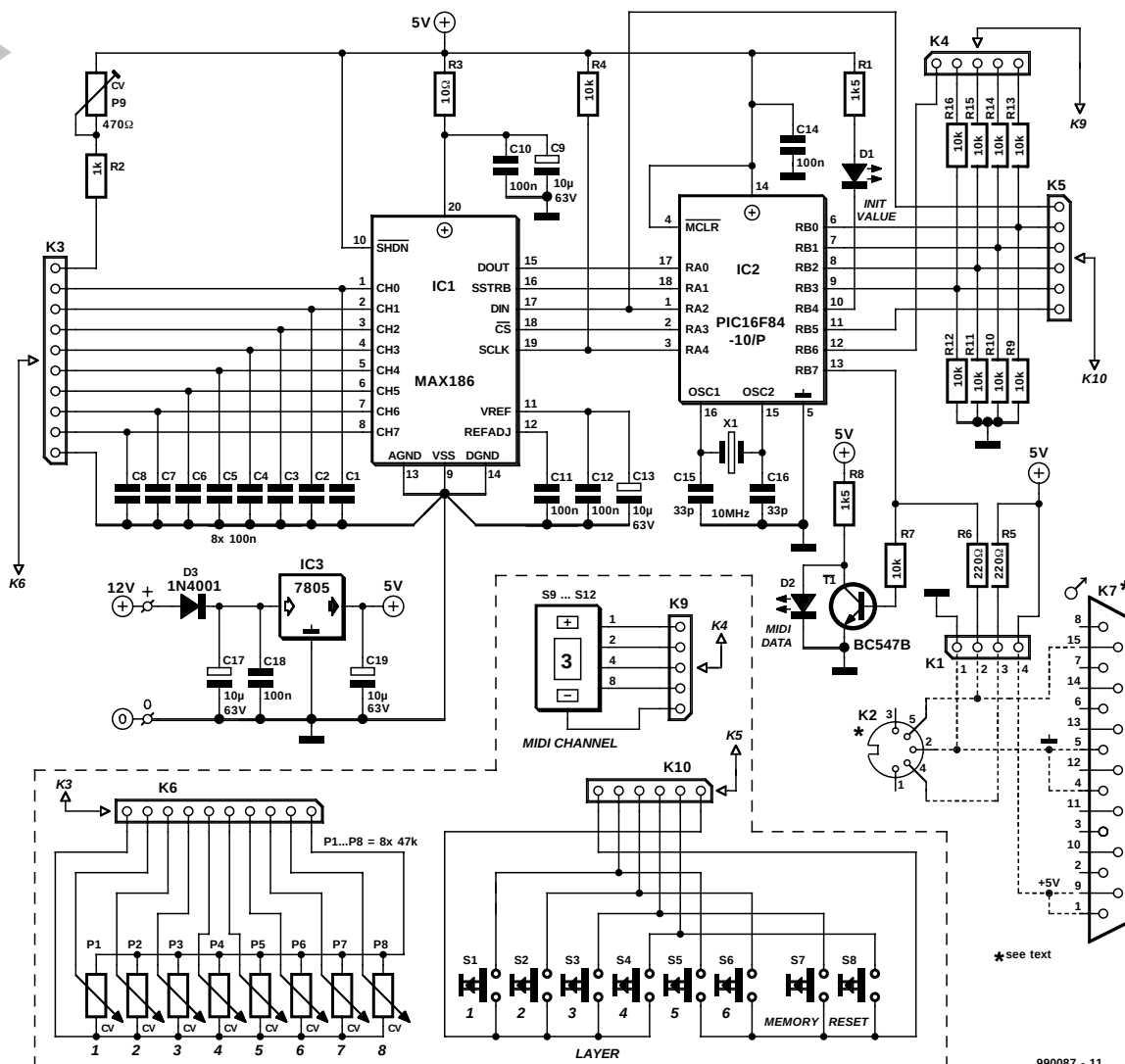
The circuit described in this article

makes working with a software synthesizer considerably easier. It utilises the ability of a software synthesizer to receive MIDI codes and to use these codes to drive certain controllers. The hardware of the MIDI parameter box can thus be kept very simple. A microcontroller reads the positions of eight standard potentiometers in turn, via an 8-channel A/D converter. If one or more of the potentiometer positions is changed, the microcontroller sends this information in MIDI format to the MIDI input of the sound card. The software synthesizer translates the MIDI codes into new settings for the controller in question.

LAYERS AND SUPER-LAYERS

The somewhat nebulous term 'layer' often crops up in connection with MIDI. Layers are actually nothing more than groups of eight synthesizer functions, which correspond to the eight potentiometers of the MIDI parameter box using a sort of multiple allocation.

Design by T. Klose



990087 - 11

Figure 1. The microcontroller reads the status of the switches and of each of the eight potentiometers in turn.

The parameter box has pushbutton switches that can select one of six layers. The MIDI codes sent by each potentiometer (or the controller) thus vary according to the layer that is selected.

The assignment of layers to specific functions is not the same for all controllers. Instead, controllers can be divided into different groups. The various types of assignments are referred to as super-layers. The MIDI parameter box knows the three most commonly used groups and adapts the MIDI codes to their specific needs. **Table 1** summarises the contents of the six layers within the three super-layers.

HARDWARE

The most important elements of the hardware, which have already been mentioned, can easily be recognised in the schematic diagram shown in **Figure 1**. The eight potentiometers are connected to channels 0 through 7 of the A/D converter IC1. The MAX186 IC, which has already been used in a

number of Elektor projects, has an interface to the microcontroller (IC2). This interface carries the output data from the D/A converter (DOUT) and the clock (SCLK), as well as the settings for the multiplexer (DIN). The converter is controlled via the SSRB and CS leads, synchronous to SCLK.

The microcontroller is a type PIC16F84 IC that is clocked at 10 MHz. In addition to the potentiometer positions, it also reads the layer selection switches (S1–S6), the MIDI channel switches (S9–S12) and two other pushbutton switches, MEMO and RESET. **Table 2** describes the meanings and uses of all of the switches.

MIDI communication with the sound card takes place via port lead RB7. The MIDI signal can be visually checked via the (blinking) LED D2. There is a good reason why two connection options are shown in the drawing. Actually, the MIDI parameter box should only be connected to the sound card via a true, optically isolated MIDI interface. Such an interface is

part of the AWE-64 package, for instance, but it cannot be ordered separately. You can either buy one for around £15 in a computer shop or copy one of the numerous Elektor designs (such as the MIDI interface in the 1995 Summer Circuits issue). A less elegant option, but one that can conceivably be used if no other MIDI devices are connected, is to connect the MIDI parameter box directly to the 15-pin joystick interface, which also has a MIDI input. The 220Ω resistor in the data line protects against short circuits if this alternative is used. This type of connection has one advantage, which is that the operating power can be drawn from the joystick port, so that D3, C17, C18 and IC3 are not needed.

This brings us to the power supply. An external power supply is obligatory with 'real' MIDI interfaces, in order to ensure the electrical isolation of the PC and the MIDI equipment. Only capacitors C17 through C19 and the voltage regulator IC3 are needed to provide a sufficiently stable +5 V. Power can be

Table 1. Contents of the six layers within the three super-layers.								
Layer	P1	P2	P3	P4	P5	P6	P7	P8
super-layer 1 (AWE, EMU8000 and EMU10k-1 synthesizers)								
layer 1	Controller 10 – 17							
layer 2	Controller 18 – 1F							
layer 3	volume	pan	expression	modulation	LP cutoff	LP reson.	chorus	reverb
layer 4	LFO1 delay	LFO1 freq.	LFO1 pitch	LFO1 vol.	LFO2 delay	LFO2 freq.	LFO2 pitch	LFO2 vol.
layer 5	env1 delay	env1 attack	env1 hold	env1 decay	env1 sustain	env1 release	env1 pitch	env1 cutoff
layer 6	env2 delay	env2 attack	env2 hold	env2 decay	env2 sustain	env2 release	-, -	
super-layer 2 (XG synthesizer)								
layer 1	Controller 10 – 17							
layer 2	Controller 18 – 1F							
layer 3	volume	pan	expression	modulation	portamento	reverb	chorus	variation
layer 4	attack	decay	release	vib. delay	vib. rate	vib. depth	cutoff	resonance
layer 5	pitch init	pitch attack	pitch rel.	p.r. time	vel. lim. L	vel. lim. H	-, -	
layer 6	pitch ben.	filter ben.	amplitude ben.	LFO PMOD ben.	FMOD D ben.	AMOD		
super-layer 3 (software synthesizer)								
layer 1	Controller 00 – 07							
layer 2	Controller 08 – 0F							
layer 3	Controller 10 – 17							
layer 4	Controller 18 – 1F							
layer 5	Controller 20 – 27							
layer 6	Controller 28 – 2F							

provided by a simple 12 V mains adapter. Diode D2 provides protection against a reverse-polarity connection.

AS SMALL AS POSSIBLE

To make the construction of the MIDI parameter box as convenient as possible, we have designed a printed circuit

board that is the size of a pack of cigarettes, as shown in **Figure 2**. It is available from Readers Services (order number **990087-1**). You shouldn't experience any problems mounting all the components, which takes around half an hour. Pay attention to the orientation of the electrolytic capacitors, the

LEDs and the ICs, and don't forget the wire bridge next to K5. After this you can prepare the control panel. The potentiometers and switches can be glued to the front panel in a reasonable arrangement, such as that shown in **Figure 3**, and wired using flatcable. You can also mount these components

Table 2. Meanings and uses of all switches.		
Component	Designation	Meaning
LED D2	MIDI Message	Message Blinks when a MIDI message is sent via the MIDO OUT port. Also blinks when a potentiometer is set exactly between two quantization levels.
LED D1	Init Value	Blinks after the power is switched on to indicate that one of the three super-layers must be selected using switches S1 – S3. Illuminated when the initial value is set for the potentiometer that was last rotated.
Switches S1 – S4	MIDI-Channel	These four binary-coded switches select the MIDI channel.
Potentiometers P1 – P8	Fader	These potentiometers are used to set the MIDI values.
Pushbuttons S1 – S6	Layer	These switches select layers 1 through 6.
Pushbutton S7	Memo	Save the last modified value in the current layer.
Pushbutton S8	Reset	Overwrite the current value with the predefined initialization value.

COMPONENTS LIST

Resistors:

R1,R8 = 1kΩ
 R2 = 1kΩ
 R3 = 10Ω
 R4,R7,R9-R16 = 10kΩ
 R5,R6 = 220Ω
 P1-P8 = 47kΩ linear potentiometer
 P9 = 470Ω preset Hi

Capacitors:

C1-C8,C10,C11,C12,C14,C18 = 100nF
 C9,C13,C17,C19 = 10μF 63V radial
 C15,C16 = 33pF

Semiconductors:

D1,D2 = LED, high efficiency
 D3 = 1N4001
 T1 = BC547B
 IC1 = MAX186BEPP
 IC2 = PIC16F84-10/P (programmed,
 order code **996521-1**)
 IC3 = 7805

Miscellaneous:

K1 = 4-way SIL connector
 K2 = 2 PCB solder pins
 K3,K6 = 10-way SIL connector
 K4,K9 = 5-way SIL connector
 K5,K10 = 6-way SIL connector
 K7 = 15-way Sub-D plug, chassis
 mount (see text)
 K8 = 5-way DIN socket, chassis
 mount, 180° (see text)
 K11 = mains adaptor socket
 S1-S8 = pushbutton, 1 make contact
 S9-S12 = on/off switch
 X1 = 10MHz quartz crystal
 Enclosure, e.g. Teko 363
 (216x130x77mm)
 PCB, order code **990087-1**

2

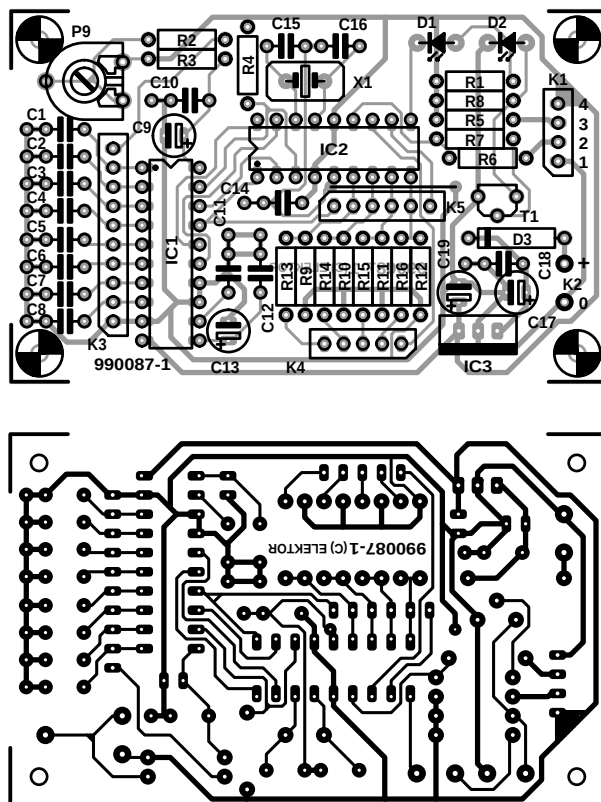


Figure 2. The MIDI parameter box can be built using this small printed circuit board.

monitor should display control codes. These will have values ranging from 0 to 127.

If this does not happen, thoroughly check the circuit construction, the cabling and the settings of the MIDI monitor. If this doesn't help, you can curse Windows or the sound card.

However, if the MIDI monitor displays the first MIDI events, then everything is in order. Trimpot P9, by the way, can also be used for calibration to adjust actual range of the MIDI values to 0 through 127.

Verify that the MIDI channel is

changed by S9-S12 (binary), and that changing the layer works properly.

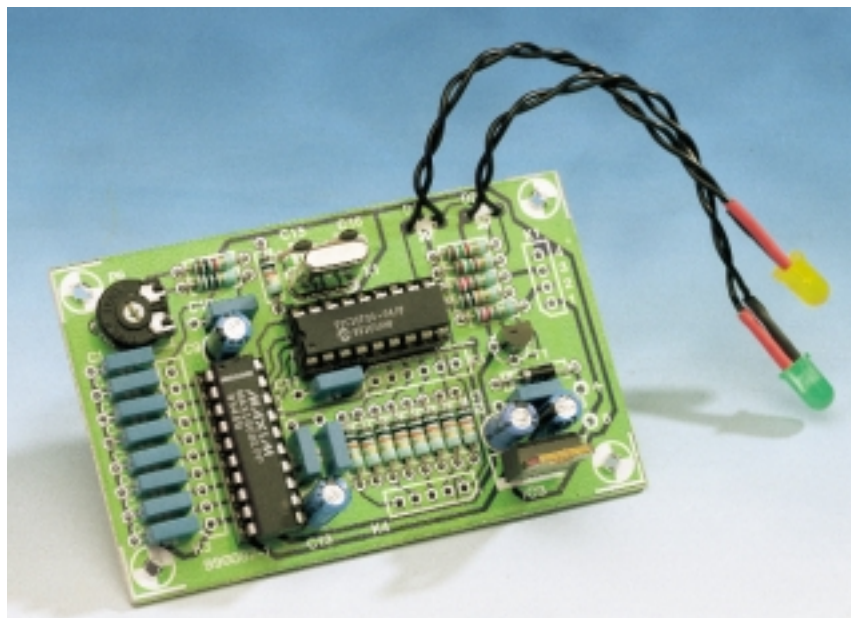
When the layer is changed, the last stored values for the potentiometers are always output via the MIDI interface. The advantage of this is that the parameters of the synthesizer or the sound card are reset to their last stored values. If for example you change the volume in layer 3, change to a different layer and some time later return to layer 3, the volume will be restored to its original level. If you want to avoid this, all you have to do is to press the Memo button before

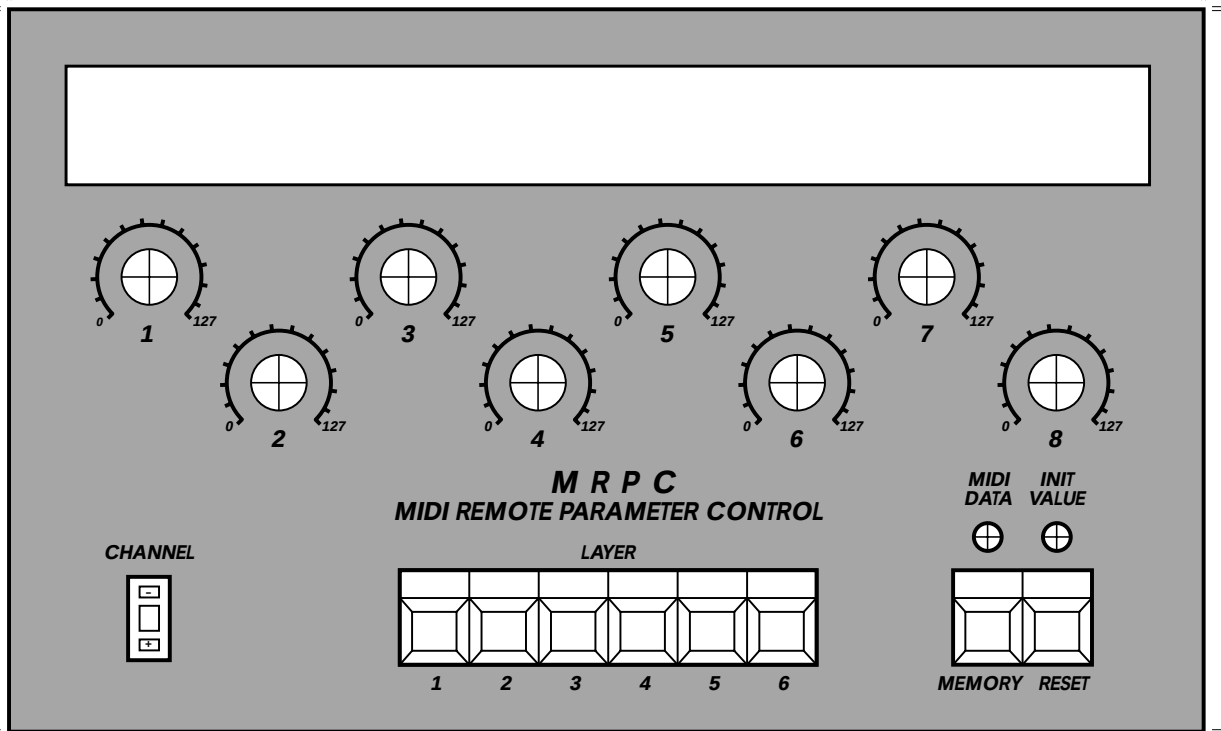
on a piece of prototyping board and wire them point-to-point. Of course, you can also design a 'real' circuit board. You should dress the flat cables such that the unit can later be built into an enclosure.

TESTING

In order to thoroughly test the MIDI parameter box, you absolutely need a MIDI monitor with a MIDI-through option for the PC, so that you can observe the transmitted MIDI data on the monitor and properly calibrate the potentiometers. The text box contains more information about suitable MIDI monitors. After a visual inspection of the soldering, connect the parts together and cable the unit to the PC. Then switch everything on and start the MIDI monitor. All the stored values in a virgin PIC are set to FFh, so they must be set to valid MIDI protocol values by pressing the Reset button. Next select super-layer 1 and layer 1 (the default layer) by pressing S1 twice.

Now comes the moment of truth. When the potentiometers are rotated, LED D2 should flash and the MIDI





990087 - F

Figure 3. A reasonable arrangement for the control elements on the front panel.

changing the layer. In this case only the values that have changed since the last layer change are stored.

(990087-1)

Text editing

(German original): R. Gerstendorf

Design editing: K. Walraven

MIDI monitors

There is a whole series of MIDI monitors that can be used with a PC. An outstandingly suitable program is MIDI-OX, for which a beta version is available for free on the Internet at www.members.xoom.com/_XOOM/MIDIOX/moxbeta.htm. After installing and starting the program, you must first select the MIDI devices, either via the menu Options/MIDI Devices or by clicking on the dark blue button with the five-pin DIN connector (see **Figure A**). On this PC, the MIDI input and output of the SoundBlaster SB16 are active.

The MIDI Port Activity window shown in **Figure B** appears if you press the bright green DIN icon in the second group of buttons. Each MIDI input and output gets its own row of 'LEDs', so that it is clear which channel is active.

The content of the transmitted data appears in the Monitors Output window. The first column shows the time when the MIDI message occurred (as noted by MIDI-OX), and the second column indicates the MIDI port via which the message arrived (in this case, Port 1 via SB16 MIDI). The following byte, 0BFh, consists of two parts: a Control Change (indicated by the 'B') and the MIDI channel number (indicated by the 'F', which corresponds to MIDI channel 16). DATA1 shows the controller number (0 – 127 in this case) and DATA2 shows the assigned value. Just as with DATA1, only the lower seven bits are used, so that the values range from 0 to 127. CHAN shows the MIDI channel once again. Note that MIDI officially uses channel numbers 1 through 16, but many programs display 0 through 15. When a different super-layer is active or the layer is changed, the messages displayed on the monitor also change. If you want to learn more background information regarding MIDI, you can find an adequate amount of literature on the Internet. One example is Eddies Home – MIDI-RPN and NRPN (<http://members.delosnet.com.tlc/nrpn.htm>).

MIDI-OX is especially well suited to checking equipment functions. If you want to go deeper into the matter, you can try easy-to-use and powerful programs such as Generator from Native Instruments or Rebirth from Propellerhead. Free demo versions of both programs are available. These can be used for only a very short time and have no save functions, but they are an outstanding choice for just playing around with the MIDI parameter box.

